Sf

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/826,427	04/03/2001	Didier Cavalli	00GR37854252	3614

7590 04/01/2004

CHRISTOPHER F. REGAN
Allen, Dyer, Doppelt, Milbrath & Gilchrist, P.A.
P.O. Box 3791
Orlando, FL 32802-3791

EXAMINER	
O BRIEN, BARRY J	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 04/01/2004 5

Please find below and/or attached an Office communication concerning this application or proceeding.

Sf

Office Action Summary	**Application No.** 09/826,427	**Applicant(s)** CAVALLI, DIDIER
	Examiner Barry J. O'Brien	**Art Unit** 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *4/03/01, 7/09/01, and 10/01/01*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Declaration and Fee as received on 7/09/2001 and Request for Correction of Filing Receipt as received on 10/01/2001.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

4. The applicant is requested to review the specification and update the status of all co-pending applications made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate.

Claim Objections

5. Claims 15-16 are objected to because of the following informalities:

   a. Claim 15 recites the limitation, "A microprocessor according to Claim 11" on its first line. Please correct the claim language to read, "The microprocessor according to Claim 11" to adhere to conventions laid out in the rest of the claims.

b. Claim 16 recites the limitation, "A microprocessor according to Claim 11" on its first line. Please correct the claim language to read, "The microprocessor according to Claim 11" to adhere to conventions laid out in the rest of the claims.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 7-10 and 17-20 are rejected under 35 U.S.C. 102(b) as being anticipated by Miu et al., U.S. Patent No. 4,484,271.

8. Regarding claim 7, Miu has taught a method for processing an interrupt request using a microprocessor executing a program, the method comprising:

a. Detecting the interrupt signal (see Col.30 lines 21-33),

b. Storing contextual data of the program (see Fig.31, Col.30 lines 50-58 and Col.45 lines 6-35),

c. Sending an interrupt acknowledge signal (see Col.30 lines 29-66 and see Col.77 line 52 – Col.78 line 26),

d. Verifying that the interrupt request is present (see Col.77 line 52 – Col.78 line 26). Here, the context was saved during the processing of the first interrupt request (see Col.30 lines 50-58), and after the first interrupt request is processed, a check is performed (verification) to see if the interrupt request signal is still provided so that further interrupts can be processed without returning to the originally interrupted program (see Col.78 lines 22-26).

e. Resuming execution of the program if the presence of the interrupt request is not verified (see Col.2 lines 12-55 and Col.78 lines 15-22). Here, if the interrupt request signal is not present after processing the first interrupt (i.e. no further interrupts are pending), control is returned to the originally interrupted program (see Col.78 lines 15-22).

f. Switching to an interrupt subroutine if the presence of the interrupt request is verified (see Col.77 line 52 – Col.78 line 26). Here, if the interrupt request signal is present after processing the first interrupt, the second interrupt is acknowledged and its corresponding ISR is executed.

9. Regarding claim 8, Miu has taught the method according to claim 7, wherein storing contextual data comprises storing data located in registers of the microprocessor in a random-access memory (see Fig.31, Col.30 lines 50-58, Col.45 lines 6-35, Col.46 lines 1-5 and Col.93 lines 50-53).

10. Regarding claim 9, Miu has taught the method according to claim 8, wherein the data stored in the random-access memory are stored into respective original registers before resuming execution of the program (see Col.47 lines 31-63).

11. Regarding claim 10, Miu has taught the method according to claim 9, wherein resuming execution of the program is postponed if a new interrupt is detected after the presence of the

interrupt request is not verified, and further comprising processing the new interrupt request (see Col.77 line 52 – Col.78 line 26).

12. Regarding claim 17, Miu has taught a microprocessor comprising:

a. A central processing unit (CPU) (see Col.30 lines 21-24) for:

I. Detecting an interrupt request during execution of a program (see Col.30 lines 21-33),

II. Storing contextual data of the program being executed (see Fig.31, Col.30 lines 50-58 and Col.45 lines 6-35),

III. Sending an interrupt acknowledge signal and switching to an interrupt subroutine if the interrupt request is present after storing the contextual data (see Col.77 line 52 – Col.78 line 26). Here, the context was saved during the processing of the first interrupt request (see Col.30 lines 50-58), and after the first interrupt request is processed, a check is performed (verification) to see if the interrupt request signal is still provided so that further interrupts can be processed without returning to the originally interrupted program (see Col.78 lines 22-26).

IV. Resuming execution of the program if the interrupt request is not present after storing the contextual data (see Col.2 lines 12-55 and Col.78 lines 15-22). Here, if the interrupt request signal is not present after processing the first interrupt (i.e. no further interrupts are pending), control is returned to the originally interrupted program (see Col.78 lines 15-22).

13. Regarding claim 18, Miu has taught the microprocessor according to claim 17, further comprising a random-access memory and registers for storing data elements, and wherein said CPU stores the contextual data by storing the data elements in said random-access memory (see Fig.31, Col.30 lines 50-58, Col.45 lines 6-35, Col.46 lines 1-5 and Col.93 lines 50-53).

14. Regarding claim 19, Miu has taught the microprocessor according to claim 18, wherein said CPU stores the data elements stored in the random-access memory in respective original registers thereof before resuming execution of the program (see Col.47 lines 31-63).

15. Regarding claim 20, Miu has taught the microprocessor according to claim 17, wherein said CPU processes a new interrupt request if the new interrupt request is present after the interrupt request is not present and before resuming execution of the program (see Col.77 line 52 – Col.78 line 26).

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 1-6 and 11-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Miu et al., U.S. Patent No. 4,484,271 (hereinafter "Miu"), in view of Short, *Embedded Microprocessor Systems Design: An Introduction Using the Intel 80C188EB* (hereinafter "Short").

18. Regarding claim 1, Miu has taught a method for processing an interrupt signal using a microprocessor comprising a central processing unit (CPU), an interrupt controller providing an interrupt request signal to the CPU when the interrupt signal is received by the microprocessor (see Col.30 lines 21-24), registers having a context stored therein corresponding to a program being executed by the CPU, and an interrupt save area for storing the context while an interrupt is being executed, the method comprising (see Fig.31 and Col.45 lines 6-35):

a. Detecting a receipt of the interrupt request signal by the CPU from the interrupt controller (see Col.30 lines 21-33),

b. Storing the context from the registers to the interrupt save area (see Fig.31, Col.30 lines 50-58 and Col.45 lines 6-35),

c. Verifying that the interrupt request signal is provided to the CPU from the interrupt controller after storing the context to the interrupt save area (see Col.77 line 52 – Col.78 line 26). Here, the context was saved during the processing of the first interrupt request (see Col.30 lines 50-58), and after the first interrupt request is processed, a check is performed (verification) to see if the interrupt request signal is still provided so that further interrupts can be processed without returning to the originally interrupted program (see Col.78 lines 22-26).

d. Sending an interrupt acknowledge signal and reading and executing a first instruction of an interrupt subroutine using the CPU if the interrupt request signal is provided to the CPU (see Col.77 line 52 – Col.78 line 26). Here, if the interrupt request signal is present after processing the first interrupt, the second interrupt is acknowledged and its corresponding ISR is executed.

e. Restoring the stored context from the interrupt save area to the registers and returning the CPU to an initial state if the interrupt request signal is not provided to the CPU (see Col.2 lines 12-55 and Col.78 lines 15-22). Here, if the interrupt request signal is not present after processing the first interrupt (i.e. no further interrupts are pending), control is returned to the originally interrupted program (see Col.78 lines 15-22).

19. Miu has not explicitly taught the interrupt save area being a stack. However, Short has taught the saving of a task's context on a when executing a context switch due to a pending interrupt so that the task can be resumed following processing of the interrupt (see Short p.468-469). One of ordinary skill in the art would have recognized that it is a requirement for correct operation that a task's context be saved so that execution can resume where it left off following processing of an interrupt. Therefore, one of ordinary skill in the art would have found it obvious to modify the interrupt save area of Miu to be a stack so that execution can be resumed from where it left off, providing correct execution results.

20. Regarding claim 2, Miu in view of Short has taught the method according to claim 1, wherein reading and executing the first instruction of the interrupt subroutine comprises reading a data element from an address determined based upon an interrupt vector provided by the interrupt controller, the data element comprising a read address of the first instruction of the interrupt subroutine (see Miu Col.45 lines 6-30).

21. Regarding claim 3, Miu in view of Short has taught the method according to claim 1, wherein returning the CPU to the initial state comprises reading an executing an instruction of

the program being executed by the CPU when the interrupt request signal was detected (see Miu Col.2 lines 12-55 and Col.78 lines 15-22).

22. Regarding claim 4, Miu in view of Short has taught the method according to claim 1, wherein returning the CPU to the initial state comprises processing a new interrupt if a new interrupt request signal is provided by the interrupt controller (see Miu Col.77 line 52 – Col.78 line 26).

23. Regarding claim 5, Miu in view of Short has taught the method according to claim 1, wherein restoring the stored context from the stack comprises restoring contents of a program counter register (see Miu Col.45 lines 6-30 and Col.78 lines 15-22, as well as Short p.469).

24. Regarding claim 6, Miu has taught the method according to claim 5, wherein returning the CPU to the initial state comprises providing the restored contents of the program counter register to an address bus (see Miu Col.45 lines 6-30 and Col.78 lines 15-22, as well as Short p.469). Here, the content of the program counter is an address of memory from which execution takes place. Therefore, when the program counter fetches the next instruction to execute, that address is supplied on a bus, which can be considered an address bus.

25. Regarding claim 11, Miu has taught a microprocessor comprising:

a. Registers having a context stored therein (see Col.45 lines 6-35 and Col.46 lines 1-25),

b. An interrupt save area for storing the context (see Col.45 lines 6-35),

c. An interrupt controller for providing an interrupt request and an interrupt vector when an interrupt signal is applied to the microprocessor (see Col.30 lines 21-24),

d. A CPU for receiving the interrupt request and the interrupt vector (see Col.30 lines 21-24), and upon detection of the interrupt request:

I. Storing the context in said interrupt save area (see Fig.31, Col.30 lines 50-58 and Col.45 lines 6-35),

II. Verifying that the interrupt request is present after storing the context (see Col.77 line 52 – Col.78 line 26). Here, the context was saved during the processing of the first interrupt request (see Col.30 lines 50-58), and after the first interrupt request is processed, a check is performed (verification) to see if the interrupt request signal is still provided so that further interrupts can be processed without returning to the originally interrupted program (see Col.78 lines 22-26).

III. Sending an interrupt acknowledge signal, and reading and executing a first instruction of an interrupt subroutine if the presence of the interrupt request is verified (see Col.77 line 52 – Col.78 line 26). Here, if the interrupt request signal is present after processing the first interrupt, the second interrupt is acknowledged and its corresponding ISR is executed.

IV. Restoring the stored context from said interrupt save area and returning the microprocessor to an initial state if the presence of the interrupt request is not verified (see Col.2 lines 12-55 and Col.78 lines 15-22). Here, if the interrupt request signal is not present after processing the first interrupt (i.e. no further interrupts are pending), control is returned to the originally interrupted program (see Col.78 lines 15-22).

26. Miu has not explicitly taught the interrupt save area being a stack. However, Short has taught the saving of a task's context on a when executing a context switch due to a pending interrupt so that the task can be resumed following processing of the interrupt (see Short p.468-469). One of ordinary skill in the art would have recognized that it is a requirement for correct operation that a task's context be saved so that execution can resume where it left off following processing of an interrupt. Therefore, one of ordinary skill in the art would have found it obvious to modify the interrupt save area of Miu to be a stack so that execution can be resumed from where it left off, providing correct execution results.

27. Regarding claim 12, Miu in view of Short has taught a microprocessor according to claim 11, wherein said CPU reads a data element from an address determined based upon the interrupt vector, the data element comprising a read address of the first instruction of the interrupt subroutine (see Miu Col.45 lines 6-30).

28. Regarding claim 13, Miu in view of Short has taught the microprocessor according to claim 11, further comprising a program counter register, and wherein said CPU restores contents of said program counter register during restoring of the stored context (see Miu Col.45 lines 6-35, Col.46 lines 48-51 and Short p.469).

29. Regarding claim 14, Miu in view of Short has taught the microprocessor according to claim 13, wherein during returning the microprocessor to the initial state said CPU reads and executes an instruction corresponding to an address in said program counter register (see Miu Col.45 lines 6-35, Col.46 lines 48-51, Col.78 lines 15-22 and Short p.469).

30. Regarding claim 15, Miu in view of Short has taught the microprocessor according to claim 11, wherein during returning the microprocessor to an initial state said CPU processes a

new interrupt if a new interrupt request is provided by said interrupt controller (see Miu Col.77 line 52 – Col.78 line 26).

31. Regarding claim 16, Miu in view of Short has taught the microprocessor according to claim 11, further comprising:

a. A 16-bit program counter register comprising two 8-bit registers (see Miu Fig.5 and Col.12 lines 42-56).

32. Miu in view of Short has not explicitly taught a 16-bit stack pointer register comprising two 8-bit registers.

33. However, Short has taught at a stack pointer contained in a 16-bit register comprising two 8-bit registers (see Short p.224) in order to keep track of the current stack scope for correctly addressing data stored on the stack, as well as allowing the implementation of a stack in ordinary RAM (see Short p.219). One of ordinary skill in the art would have recognized that one couldn't arbitrarily store data in memory without order or a way to retrieve the data, and that without a way to retrieve the correct data, incorrect operation of the processor can occur. Therefore, one of ordinary skill in the art would have found it obvious to modify the processor of Miu to include a 16-bit stack pointer register comprising two 8-bit registers so that context switch data on the stack can be retrieved correctly.

Conclusion

34. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the

references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

35. Hu, U.S. Patent No. 5,822,595, has taught a method for processing interrupts using an in-progress priority stack to keep track of priorities.

36. Futral, U.S. Patent No. 6,708,241, has taught a method for processing interrupts such that spurious interrupts are avoided.

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Barry J. O'Brien whose telephone number is (703) 305-5864. The examiner can normally be reached on Mon.-Fri. 6:30am-4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

38. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Barry J. O'Brien
Examiner
Art Unit 2183

BJO
3/29/2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100